Team 29: Crime Data Management System

Maria Andrea Ona
Nicola Bini
MSBA1

Introduction

Definition of crime
"The National UCR Program defines an incident as one or more offenses committed by the same offender, or group of offenders acting in concert, at the same time and place." [1] In this document, term incident(s) will be referred as "crime".

The hierarchy rule
Generally, if a series of offenses are committed in the same event, one record should be made for each offense committed.[1] However, for statistical purposes, if multiple offenses are committed within the same crime, the crime is stored and classified as the most serious of the offenses. In this way, crimes with multiple offenses are not counted more than once and the understanding of statistical data is facilitated. For example, statistics for "larceny" will include crimes where larceny is the primary offense. If a violent offense is committed during a larceny, then the crime will be classified as "robbery". Since our database is for statistical purposes only, when a crime occurs, we will only store the most serious offense.

Users

- Police Officers
- Court Officials
- City Managers and Officials
- National Policy Makers

Purpose and Significance

This data management system is primarily designed for the above-mentioned authorities to:

Police Officers: Understand in which parts of the city crimes are more likely to occur and streamline the law enforcement's efforts effectively by focusing on high-risk areas and organize patrols accordingly to criminal activities.

Court Officials: Store information resulting from court trials for statistical purposes

City Managers: Understand the current level of criminality in the city, acknowledge what are the most pressing criminality issues impacting the residents and use such information to take decision about what social investments the city should make. Inform current and prospective residents, entrepreneurs, investors and stakeholders of the safety of the city and specific neighborhoods.

National Policy makers: Compare different criminality levels in different cities of the country to assess which factors are stimulating crimes and reinstate the existing policies in mitigating the criminality. Also, this can be a useful reference as to what policies should be made next. With this, the authorities can decide which neighborhoods need social investments and policies the most. Evaluate whether the convicted criminals have been able to readjust or they have perpetrated crimes after attending corrective programs. Highlight emerging social problems (e.g., juvenile criminality) and promptly address them.

Reports

Police Officers are concerned about improving their efficacy in reducing criminal activity in the city. Using reports concerning **number of crimes per neighborhood** and **number of crimes per type** they would be able to better streamline their tasks more effectively.

Court Officials will only access the database for inserting data without generating reports.

City Managers want to identify problematic neighborhoods by looking reports such as **number of crimes committed per neighborhood** and **number of offenders living in a specific area**. They may want to generate report about **volume of criminal activities between their city and a similar city** per dimension or socio-economic structure.

National Policy makers are interested in reports concerning national trends, such as **number of violent or domestic crimes in United States over years**, or they want **to compare criminal activities in different cities.**

Scope and Limitation

In reference to the purpose of the database, the information requires the following (limited within close proximity):

- Crime location (where the crimes happened)
- Type of number of crimes committed in a given city or by a given person
- Residence of perpetrators of crimes
- Date and time of crime
- Convicted criminals' crime history
- Criminals personal information
- Classification of crimes into domestic/non-domestic and violent/non-violent

All information that is not relevant for statistical will not be stored. In particular, the database does not contain:

- Info related to the court and trial process
- Info related to solving crimes (evidence)

Sources:
[1] *https://www.fbi.gov/file-repository/ucr/ucr-2019-1-nibrs-user-manua-093020.pdf/view*

1. Flow of System

i. Reporting a new crime
Upon report of the crime, the appointed police officer can submit a new record indicating the crime name, date and time of the crime, the address of where the crime happened, etc. The officer may leave some fields blank if the information is not available as of the moment. Once the officer has completely secured all information the database requires, he can then compile this information in the database by simply modifying or appending the fields.

ii. An offender was convicted
When an offender has historical background of being convicted from previous crimes, (excluding the recent one) court officials may update the latest crime record with all the information resulting from the judicial decision that may transpire. To detail, if a crime is reported and recorded as a robbery but the convicted criminal is charged with larceny only, then the crime (offense) name will be changed from "Larceny" to "Robbery". Additionally, court

officials may add all newly acquired information related to victims, offenders, residencies of all the people involved, date, time and location of the crime.

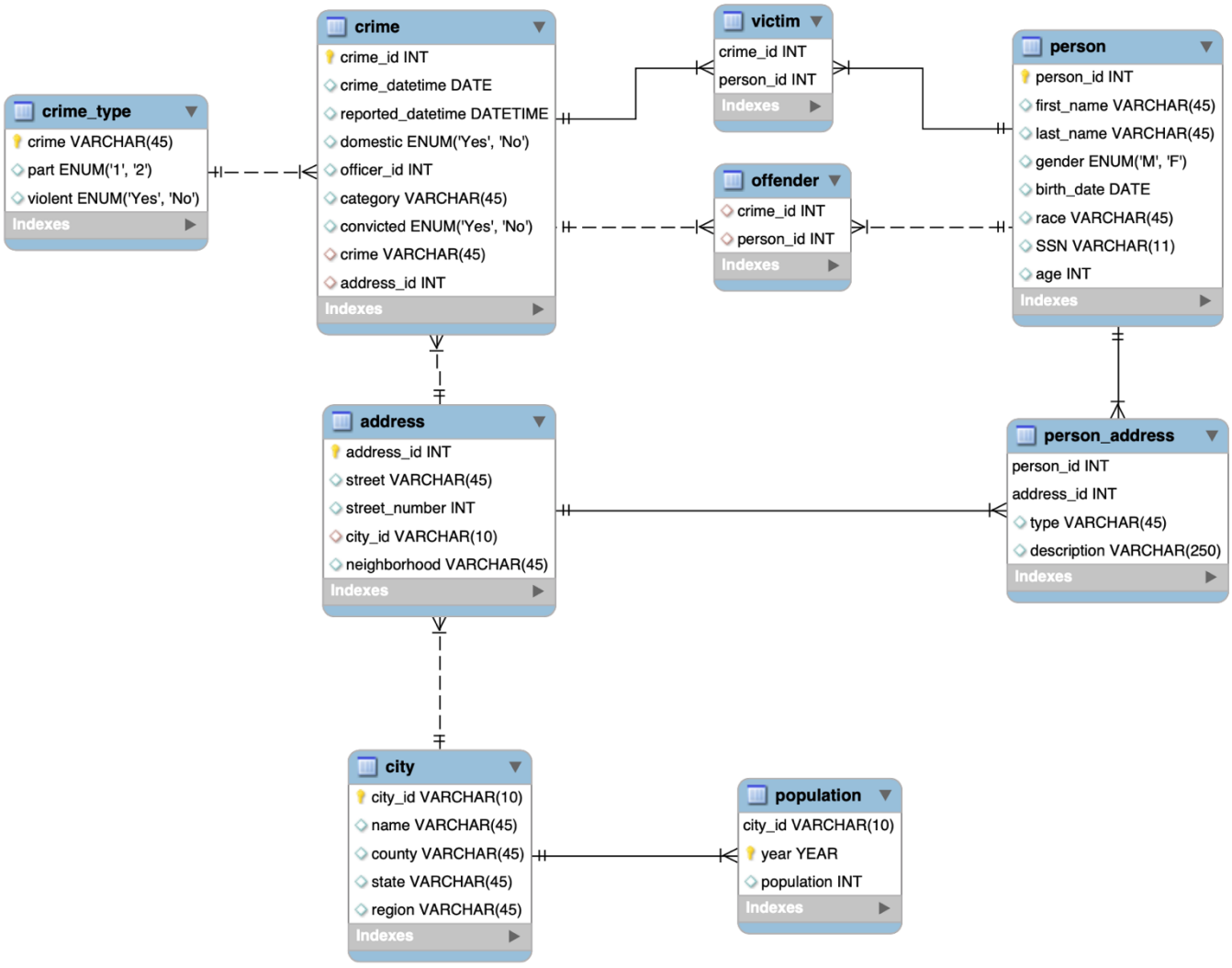
iii. Updating population data
The Census Bureau collects and reports data about the population of every US city for each year. Once every year, when new statistical data population data becomes available, the MySQL Server automatically connects to the Census Bureau's database to create a new record for each city with the population of that year.

iv. Accessing data
The access of personal information of victims and offenders is restricted based on the law of each state. Generally, politicians and city officials will not have access to personal information of victims and offenders. Personal information of victims and offenders will be accessible only by authorized police officers and court officials to verify the correctness of the data.

2. Database structure

Table Name	Primary Key(s)	Foreign Key(s)	Description
crime	crime_id	address_id, crime	Crime master table, it contains all necessary information/details about the cases filed by a precinct officer, including timestamps and classification of charges.
person	person_id		Contains record of personal data of the people involved in crimes (both victims and offenders)
city	city_id		Information about the cities where crimes have been committed, or where the people involved reside
population	city_id, year	city_id	Cities population per year
victim	person_id, crime_id	person_id, crime_id	Match offenders with the crimes they have suffered from
offender	person_id, crime_id	person_id, crime_id	Match offenders with the crimes they have committed
address	address_id	city_id	Contains residential information of the people involved (victims and offenders) and locale of crimes
person_address	person_id, address_id	person_id, address_id	Match each address with the person it belongs to, specifying if the person lives or works there.
crime_type	crime		Classify each crime as Part I or Part II crime and as violent or non-violent

3. Entity-Relational (ER) Model



3. Documentation of SQL queries

1. Report that displays the number of crimes committed in the United States per year, segmented by violent and non-violent

QUERY:

```
SELECT year(c.crime_datetime) as "Year", COUNT(c.crime_id) as "Crimes in thE US",
       ct.violent as "Violent Crimes"
FROM crime_type ct, crime c
WHERE c.crime = ct.crime
GROUP BY ct.violent, Year
ORDER BY Year asc;
```

OUTPUT:

Year	Crimes in the US	Violent Crimes
1990	46	Yes
1990	94	No
1991	97	No
1991	46	Yes
1992	45	Yes
1992	115	No
1993	118	No
1993	31	Yes
1994	109	No

2. Report that displays the number of crimes per city in 2018.

QUERY:

```
SELECT cy.name as "City", COUNT(*) as "Number of Crimes"
FROM crime c, crime_type ct, address a, city cy
WHERE c.crime = ct.crime
AND c.address_id = a.address_id
AND a.city_id = cy.city_id
AND year(c.crime_datetime) = 2018
GROUP BY cy.name;
```

OUTPUT:

City	Number of Crimes
Cambridge	24
Houston	27
Boston	33
Seattle	28
San Francisco	26

3. Report that displays the number of domestic crimes that happened in the United States per year.

QUERY:

```
SELECT year(crime_datetime) as "Year", COUNT(crime_id) as "Domestic Crimes"
FROM crime
WHERE domestic = "Yes"
GROUP BY Year
ORDER BY Year asc;
```

OUTPUT:

Year	Domestic Crimes
1992	40
1993	48
1994	51
1995	37
1996	29
1997	45
1998	49
1999	42

4. Report report that displays the statistics of crimes per city.

QUERY:

```
select cy.name as "City", c.crime as "Crime", COUNT(crime_id) as "Number of Crimes"
FROM crime c, address a, city cy
WHERE c.address_id = a.address_id
AND a.city_id = cy.city_id
GROUP BY c.crime, cy.name
ORDER BY city;
```

OUTPUT:

City	Crime	Number of Crimes
Boston	Robbery	41
Boston	Hit and Run	56
Boston	Threats	58
Boston	Trespassing	53
Boston	Assault	63
Boston	Fraud	70
Boston	Sex offense	48
Boston	Drug offense	46
Boston	Auto theft	59
Boston	Kidnapping	66
Boston	Murder	72
Boston	Burglary	51
Boston	Extortion	44
Boston	Rape	53
Boston	Property de...	60
Boston	Larceny	64
Cambridge	Drug offense	48
Cambridge	Rape	59
Cambridge	Threats	51
Cambridge	Fraud	50

5. Report that displays the average number of crimes per month and orders the output by the months with the highest average of number of crimes.

QUERY:

```
CREATE OR REPLACE VIEW n_crimes_per_month AS
SELECT year(c.crime_datetime), monthname(c.crime_datetime) as Months, COUNT(c.crime_id) as Number
of Crimes
FROM crime c, address a, city cy, crime_type ct
WHERE c.address_id = a.address_id
AND c.crime = ct.crime
AND a.city_id = cy.city_id
GROUP BY monthname(c.crime_datetime), month(c.crime_datetime), year(c.crime_datetime);

SELECT Months, ROUND(AVG(`Number of Crimes`),1) AS Average number of crimes
FROM n_crimes_per_month
GROUP BY Months
ORDER BY Average number of crimes DESC;
```

OUTPUT:

Months	Average number of crim...
November	13.5
December	13.4
August	13.3
July	12.9
April	12.6
October	12.5
January	12.3
March	12.3
June	12.2
September	12.2
May	11.5
February	10.8

6. Socioeconomic factors are said to be linked to crime rates, areas with exposure to crime greatly affect its people having poor access to education, poverty, drugs abuse, etc. Generate a report that displays the number of crimes per neighborhood in all cities and orders it by cities and neighborhoods.

QUERY:

```
SELECT ct.name as "City", a.neighborhood as "Neighborhood",  COUNT(crime_id) as "Number of Crimes"
FROM crime c, address a, city ct
WHERE c.address_id = a.address_id
AND a.city_id = ct.city_id
GROUP BY a.neighborhood, ct.name
ORDER BY ct.name, a.neighborhood;
```

OUTPUT:

City	Neighborhood	Number of Crimes
Boston	Bay Village	32
Boston	Beacon Hill	47
Boston	Brighton	37
Boston	Charlestown	48
Boston	Chinatown–Leather District	39
Boston	Dorchester	46
Boston	Downtown	34
Boston	East Boston	38
Boston	Fenway-Kenmore	39
Boston	Hyde Park	43

7. In reference to the query above, display the number of crimes committed by people living in a specific neighborhood to know what neighborhoods need social assistance to prevent criminality.

QUERY:

```
SELECT ct.name as "City", a.neighborhood, COUNT(p.person_id) as "Number of Offenders"
FROM offender o, person p, crime c, address a, person_address pa, city ct
WHERE c.crime_id = o.crime_id
AND o.person_id = p.person_id
AND p.person_id = pa.person_id
AND pa.type = "Home"
AND pa.address_id = a.address_id
AND ct.city_id = a.city_id
GROUP BY ct.name, a.neighborhood;
```

OUTPUT:

City	neighborhood	Number of Offend...
Seattle	Northgate	19
Cambridge	Mid-Cambridge	40
Seattle	Bitter Lake	12
Houston	IAH Airport	9
Boston	West End	12
Cambridge	MIT	34
Seattle	North College Park	11
San Francisco	Fillmore	9
Houston	Edgebrook	6
San Francisco	Cathedral Hill	13

8. Violence against women. According to https://now.org/resource/violence-against-women-in-the-united-states-statistic/, advocacy groups are working together to halt the gender-based violence where approximately 1,200 women lives were spared. Given this, create a report that displays the number of Femicides in the US in the last 10 years, by city.

QUERY:

```
SELECT ct.name as “City”, COUNT(1) AS "Femicides"
FROM victim v, offender o, crime cr, person po, person pv, address a, city ct
WHERE cr.crime_id = o.crime_id
AND cr.crime_id = v.crime_id
AND o.person_id = po.person_id
AND v.person_id = pv.person_id
AND cr.crime = "Murder"
AND po.gender = "M"
AND pv.gender = "F"
AND cr.address_id = a.address_id
AND a.city_id = ct.city_id
AND (2020 - YEAR(cr.crime_datetime)) < 10
GROUP BY ct.name;
```

OUTPUT:

City	Femicides
San Francisco	5
Boston	5
Seattle	2
Cambridge	3

9. Given that there is a controversy concerning the race/ethnicity of murderers, create a report that displays the number of offenders that have committed murder and group them by racial groups.

QUERY:

```
SELECT c.crime as "Crime", p.race as "Race/Ethnicity", COUNT(o.crime_id) as "Number of offenders"
FROM person p, crime c, offender o, crime_type ct
WHERE p.person_id = o.person_id
AND o.crime_id = c.crime_id
AND ct.crime = c.crime
AND c.crime = "Murder"
GROUP BY p.race, c.crime;
```

OUTPUT:

Crime	Race/Ethnicity	Number of offenders
Murder	Non-Hispanic white	50
Murder	Asian	54
Murder	Native Hawaiians and Other Pacific Islanders	33
Murder	Native Americans and Alaska Natives	36
Murder	Two or more races	39
Murder	Black or African American	48
Murder	Hispanic and Latino	37

10. Create a report that shows the average age of people who have been committing drug offenses, showing the number of cases last year (2019) by city. In this case, we can determine among which age group is most likely to commit a crime?

QUERY:

```
SELECT cy.name as "City",TRUNCATE(avg(p.age), 0) as "Average age"
FROM crime c, address a, city cy, crime_type ct, offender o, person p
WHERE c.address_id = a.address_id
AND c.crime = ct.crime
AND a.city_id = cy.city_id
AND c.crime_id = o.crime_id
AND o.person_id = p.person_id
AND year(c.crime_datetime) = 2019
AND ct.crime = "Drug Offense"
GROUP BY cy.name;
```

OUTPUT:

City	Average age
Seattle	33
San Francisco	57
Houston	58
Boston	45
Cambridge	49

3. SQL Procedures

1. n_crimes_over_years

Outline: Stored procedure that provides the number of cases of a particular crime in a given city over years
Input: city_name varchar(45) and crime varchar(45)
Output: number of crimes in a given city per year

DESCRIPTION:

The procedure takes as input a crime and a city. It returns a table where with a row for each year and the number of occurrences of the specified crime.

PROCEDURE CODE:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `n_crimes_over_years`(IN in_city_name
VARCHAR(45), IN in_crime VARCHAR(45))
BEGIN

SELECT ct.name as `City`, year(cr.crime_datetime) as `Date(Year)`, COUNT(1) as `Number of
Crimes`
FROM city ct, crime cr, address a, population p
WHERE cr.address_id = a.address_id
AND a.city_id = ct.city_id
AND ct.city_id = p.city_id
AND ct.name = in_city_name
AND cr.crime = in_crime
GROUP BY p.city_id, Date (Year)
ORDER BY Date (Year);

END
```

PROCEDURE OUTPUT:

Date (Year)	Number of Crimes
1992	48
1994	12
1995	36
1996	24
1997	24
1998	12
1999	12
2000	24
2001	12
2002	12
2003	36
2004	12
2005	12

2. calc_age

Outline: Stored procedure that calculates the age of people had in a year specified by the user.
Input: year at which we are calculating the age of people (example: DATE '2012-12-12')
Output: No output, it creates or updates the column `age` in the table `person`

DESCRIPTION:

First, this procedure checks if the column age in the table person already exists, if it does not, it creates it. Then, it cycles through each person in the table `person`, calculating how old each person was at the year given as input, and saving the value in the columns 'age'.

PROCEDURE CODE:

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `calc_age`(IN in_date DATE)
BEGIN

DECLARE cur_person INT DEFAULT 0;

DECLARE cur1 CURSOR FOR
 SELECT person_id
 FROM person;

IF NOT EXISTS ( SELECT NULL
    FROM INFORMATION_SCHEMA.COLUMNS
    WHERE table_name = 'person'
    AND table_schema = 'crimedb'
    AND column_name = 'age')
  THEN ALTER TABLE person
  ADD age INT;

END IF;

OPEN cur1;

a1:LOOP
 FETCH cur1 INTO cur_person;
 UPDATE person
 -- Dividing by 365.25 instead of 365 because of leap years
 SET age = DATEDIFF(in_date, birth_date)/365.25
 WHERE person_id = cur_person;

END LOOP a1;
CLOSE cur1;

END$$
DELIMITER ;
```

PROCEDURE OUTPUT:

person_id	first_name	last_name	gender	birth_date	race	SSN	age
741	Daniel	Roswick	M	1963-11-29	Two or more races	563-36-4185	47
743	John	Wekenborg	M	1941-07-15	Black or African American	798-62-0659	69
745	Harland	Mcauley	M	1971-09-20	Hispanic and Latino	231-06-3383	39
747	Charles	Howard	M	1952-11-11	Native Hawaiians and Other Pacific Islanders	742-22-8610	58
750	Jack	Corcoran	M	1960-07-08	Black or African American	513-22-9825	50
752	Alexander	Lloyd	M	1976-10-01	Asian	278-06-0918	34
753	Clifford	Jarvis	M	1948-04-18	Black or African American	110-10-0463	63
756	Jon	Wilkerson	M	1945-07-21	Black or African American	557-18-8079	65
759	Ronald	Henderson	M	1931-02-06	Black or African American	865-06-4648	80
760	Juan	Hartwell	M	1970-01-18	Native Hawaiians and Other Pacific Islanders	526-15-6328	41
762	Kent	Haney	M	1966-10-24	Native Americans and Alaska Natives	645-78-0616	44
765	Roger	Kutz	M	1961-02-01	Native Hawaiians and Other Pacific Islanders	095-31-9014	50
767	Richard	Davis	M	1946-10-31	Native Hawaiians and Other Pacific Islanders	190-57-9602	64

Appendix

1. Create table scripts

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,E
RROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema crimedb
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema crimedb
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `crimedb` DEFAULT CHARACTER SET utf8 ;
USE `crimedb` ;

-- -----------------------------------------------------
-- Table `crimedb`.`city`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`city` (
  `city_id` VARCHAR(10) NOT NULL,
  `name` VARCHAR(45) NULL DEFAULT NULL,
  `county` VARCHAR(45) NULL DEFAULT NULL,
  `state` VARCHAR(45) NULL DEFAULT NULL,
  `region` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`city_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `crimedb`.`address`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`address` (
  `address_id` INT NOT NULL,
  `street` VARCHAR(45) NULL DEFAULT NULL,
  `street_number` INT NULL DEFAULT NULL,
  `city_id` VARCHAR(10) NULL DEFAULT NULL,
  `neighborhood` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`address_id`),
  INDEX `city_id_idx` (`city_id` ASC) VISIBLE,
  CONSTRAINT `city_id`
    FOREIGN KEY (`city_id`)
    REFERENCES `crimedb`.`city` (`city_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

```
-- -----------------------------------------------------
-- Table `crimedb`.`crime_type`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`crime_type` (
  `crime` VARCHAR(45) NOT NULL,
  `part` ENUM('1', '2') NULL DEFAULT NULL,
  `violent` ENUM('Yes', 'No') NULL DEFAULT NULL,
  PRIMARY KEY (`crime`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `crimedb`.`crime`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`crime` (
  `crime_id` INT NOT NULL,
  `crime_datetime` DATE NULL DEFAULT NULL,
  `reported_datetime` DATETIME NULL DEFAULT NULL,
  `domestic` ENUM('Yes', 'No') NULL DEFAULT NULL,
  `officer_id` INT NULL DEFAULT NULL,
  `category` VARCHAR(45) NULL DEFAULT NULL,
  `convicted` ENUM('Yes', 'No') NULL DEFAULT NULL,
  `crime` VARCHAR(45) NULL DEFAULT NULL,
  `address_id` INT NULL DEFAULT NULL,
  PRIMARY KEY (`crime_id`),
  INDEX `crime_idx` (`crime` ASC) VISIBLE,
  INDEX `address_id_idx` (`address_id` ASC) VISIBLE,
  CONSTRAINT `address_id`
    FOREIGN KEY (`address_id`)
    REFERENCES `crimedb`.`address` (`address_id`),
  CONSTRAINT `crime`
    FOREIGN KEY (`crime`)
    REFERENCES `crimedb`.`crime_type` (`crime`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `crimedb`.`person`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`person` (
  `person_id` INT NOT NULL,
  `first_name` VARCHAR(45) NULL DEFAULT NULL,
  `last_name` VARCHAR(45) NULL DEFAULT NULL,
  `gender` ENUM('M', 'F') NULL DEFAULT NULL,
  `birth_date` DATE NULL DEFAULT NULL,
  `race` VARCHAR(45) NULL DEFAULT NULL,
  `SSN` VARCHAR(11) NULL DEFAULT NULL,
  `age` INT NULL DEFAULT NULL,
  PRIMARY KEY (`person_id`))
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `crimedb`.`offender`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`offender` (
  `crime_id` INT NULL DEFAULT NULL,
  `person_id` INT NULL DEFAULT NULL,
  INDEX `person_id_idx` (`person_id` ASC) VISIBLE,
  INDEX `crijme` (`crime_id` ASC) VISIBLE,
  CONSTRAINT `crijme`
    FOREIGN KEY (`crime_id`)
    REFERENCES `crimedb`.`crime` (`crime_id`),
  CONSTRAINT `perskon`
    FOREIGN KEY (`person_id`)
    REFERENCES `crimedb`.`person` (`person_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `crimedb`.`person_address`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`person_address` (
  `person_id` INT NOT NULL,
  `address_id` INT NOT NULL,
  `type` VARCHAR(45) NULL DEFAULT NULL,
  `description` VARCHAR(250) NULL DEFAULT NULL,
  PRIMARY KEY (`person_id`, `address_id`),
  INDEX `address_id_idx` (`address_id` ASC) VISIBLE,
  CONSTRAINT `address`
    FOREIGN KEY (`address_id`)
    REFERENCES `crimedb`.`address` (`address_id`),
  CONSTRAINT `person`
    FOREIGN KEY (`person_id`)
    REFERENCES `crimedb`.`person` (`person_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `crimedb`.`population`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`population` (
  `city_id` VARCHAR(10) NOT NULL,
  `year` YEAR NOT NULL,
  `population` INT NULL DEFAULT NULL,
  PRIMARY KEY (`year`, `city_id`),
  INDEX `city` (`city_id` ASC) VISIBLE,
  CONSTRAINT `city`
    FOREIGN KEY (`city_id`)
```

```
    REFERENCES `crimedb`.`city` (`city_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `crimedb`.`victim`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `crimedb`.`victim` (
  `crime_id` INT NOT NULL,
  `person_id` INT NOT NULL,
  PRIMARY KEY (`crime_id`, `person_id`),
  INDEX `person_id_idx` (`person_id` ASC) VISIBLE,
  CONSTRAINT `crime_id`
    FOREIGN KEY (`crime_id`)
    REFERENCES `crimedb`.`crime` (`crime_id`),
  CONSTRAINT `person_id`
    FOREIGN KEY (`person_id`)
    REFERENCES `crimedb`.`person` (`person_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

2. Insert scripts

See attachment '**crimedb-schema-data.sql**` to import schema, tables and data.